K-medoids Algorithm

Cluster Analysis – Partitioning Approach

Ricco RAKOTOMALALA
Université Lumière Lyon 2

Outline

Cluster analysis

Clustering, unsupervised learning

Cluster analysis

Also called: clustering, unsupervised learning, typological analysis

Input variables, used for the creation of the clusters

Often (but not always) numeric variables

Modele	puissance	cylindree	vitesse	longueur	largeur	hauteur	poids	co2
PANDA	54	1108	150	354	159	154	860	135
TWINGO	60	1149	151	344	163	143	840	143
YARIS	65	998	155	364	166	150	880	134
CITRONC2	61	1124	158	367	166	147	932	141
CORSA	70	1248	165	384	165	144	1035	127
FIESTA	68	1399	164	392	168	144	1138	117
CLIO	100	1461	185	382	164	142	980	113
P1007	75	1360	165	374	169	161	1181	153
MODUS	113	1598	188	380	170	159	1170	163
MUSA	100	1910	179	399	170	169	1275	146
GOLF	75	1968	163	421	176	149	1217	143
MERC_A	140	1991	201	384	177	160	1340	141
AUDIA3	102	1595	185	421	177	143	1205	168
CITRONC4	138	1997	207	426	178	146	1381	142
AVENSIS	115	1995	195	463	176	148	1400	155
VECTRA	150	1910	217	460	180	146	1428	159
PASSAT	150	1781	221	471	175	147	1360	197
LAGUNA	165	1998	218	458	178	143	1320	196
MEGANECC	165	1998	225	436	178	141	1415	191
P407	136	1997	212	468	182	145	1415	194
P307CC	180	1997	225	435	176	143	1490	210
PTCRUISER	223	2429	200	429	171	154	1595	235
MONDEO	145	1999	215	474	194	143	1378	189
MAZDARX8	231	1308	235	443	177	134	1390	284
VELSATIS	150	2188	200	486	186	158	1735	188
CITRONC5	210	2496	230	475	178	148	1589	238
P607	204	2721	230	491	184	145	1723	223
MERC_E	204	3222	243	482	183	146	1735	183
ALFA 156	250	3179	250	443	175	141	1410	287
BMW530	231	2979	250	485	185	147	1495	231

Goal: Identifying the set of objects with similar characteristics

We want that:

(1) The objects in the same group are more similar to each other

(2) Thant to those in other groups

For what purpose?

→ Identify underlying structures in the data

→ Summarize behaviors or characteristics

→ Assign new individuals to groups

→ Identify totally atypical objects

The aim is to detect the set of "similar" objects, called groups or clusters.

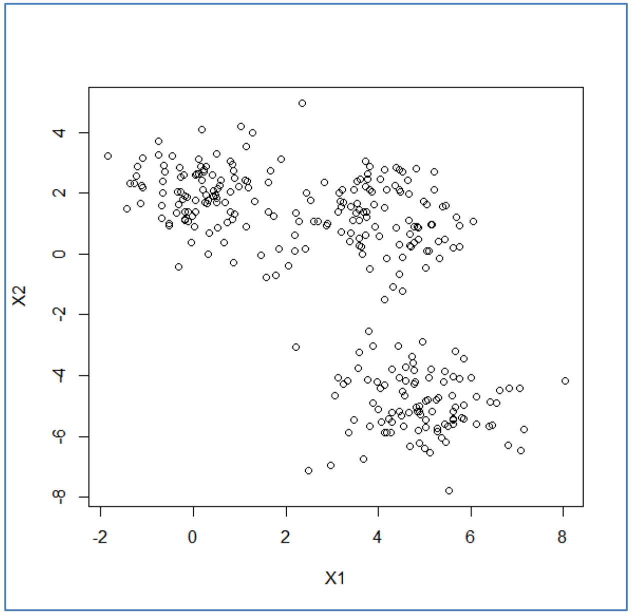
"Similar" should be understood as "which have close characteristics".

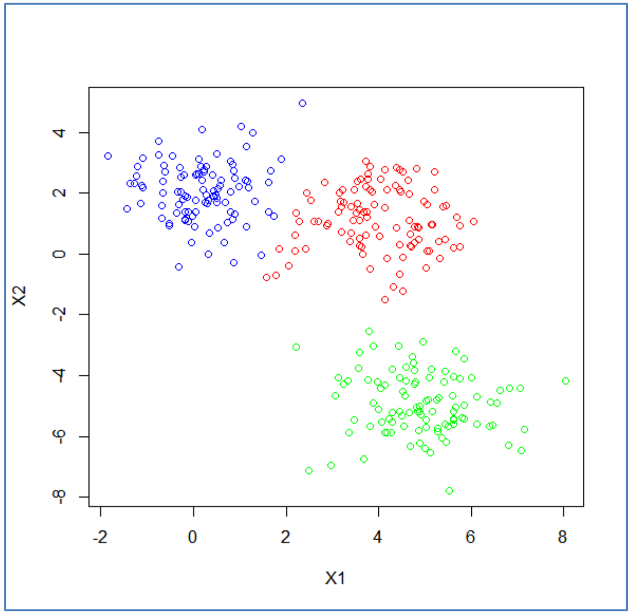
Cluster analysis

Example into a two dimensional representation space

We "perceive" the groups of instances (data points) into the representation space.

The clustering algorithm has to identify the “natural” groups (clusters) which are significantly different (distant) from each other.





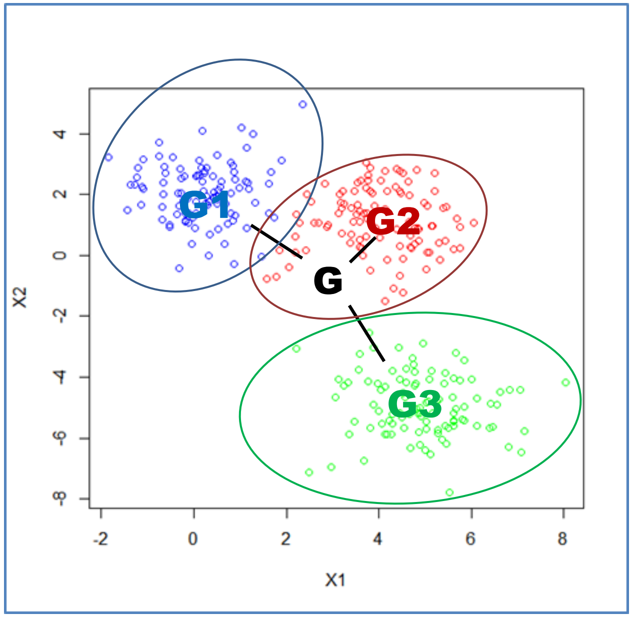
2 key issues

1. Determining the number of clusters
2. Delimiting these groups by machine learning algorithm

Characterizing the partition

Within-cluster sum of squares (variance)

Give crucial role to the centroids



Note: *Since the instances are attached to a group according to their proximity to their centroid, the shape of the clusters tends to be spherical.*

Huygens theorem

$$\text{TOTAL.SS} = \text{BETWEEN-CLUSTER.SS} + \text{WITHIN-CLUSTER.SS}$$

$$\mathrm{T} = B + W$$

$$\sum_{i=1}^{n} d^2(i,G) = \underbrace{\sum_{k=1}^{K} n_k d^2(G_k,G)}_{} + \underbrace{\sum_{k=1}^{K}\sum_{i=1}^{n_k} d^2(i,G_k)}_{}$$

Dispersion of the clusters' centroids around the overall centroid.
Clusters separability indicator.

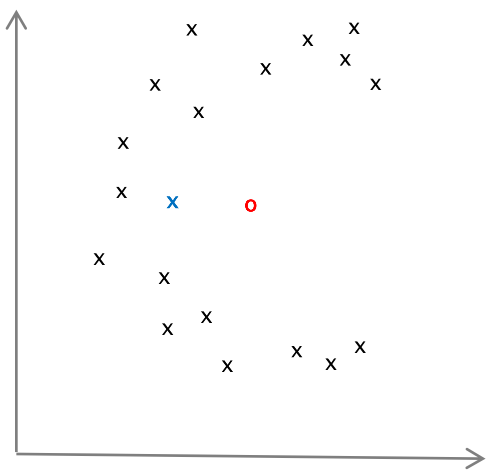
Dispersion inside the clusters.
Clusters compactness indicator.

d() is a distance measurement characterizing the proximity between individuals. E.g. Euclidean distance or Euclidean distance weighted by the inverse of variance. Pay attention to outliers.

The aim of the cluster analysis would be to minimize the within-cluster sum of squares (W), to a fixed number of clusters (e.g. K-Means algorithm).

The concept of "medoid"

Representative data point of a cluster



The centroid (o) may be totally artificial, it may not correspond to the real configuration of the dataset.

The concept of medoid (x) is more appropriate in some circumstances. This is an observed data point which minimizes its distance to all the other instances.

$$M = \arg\min_{m} \sum_{i=1}^{n} d(i,m)$$

m = 1, …, n ; each data point is candidate to be medoid.

$$E = \sum_{k=1}^{K} \sum_{i=1}^{n_k} d(i, M_k)$$

It can be used as measure for the quality of the partition, instead of the within cluster sum of squares.

$$d(i,i') = \sum_{j=1}^{p} \left| x_{ij} - x_{i'j} \right|$$

We are no longer limited to the Euclidean distance. The Manhattan distance for instance allows to dramatically reduces the influence of outliers.

Partitioning-based clustering

Generic iterative relocation clustering algorithm

Main steps

- Set the number of clusters K
- Set a first partition of the data
- **Relocation.** Move objects (instances) from one group to another to obtain a better partition
- The aim (implicitly or explicitly) is to optimize some objective function evaluating the partitioning
- Provides an unique partitioning of the objects (unique solution)

But can be depending on other parameters such as the maximum diameter of the clusters. Remains an open problem often.

Often in a random fashion. But can also start from another partition method or rely on considerations of distances between individuals (e.g., the K most distant individuals from each other).

By processing all individuals, or by attempting to have random exchanges (more or less) between groups.

The measure E will be used (see the previous slide).

We have a unique solution for a given value of K. And not a hierarchy of partitions as for HAC (hierarchical agglomerative clustering) for example.

K-medoids algorithm

Several possible approaches

K-Medoids Algorithm

A variant of K-Means Algorithm

! It is necessary to calculate the matrix of pairwise distances between individuals d(i,i'), i,i' = 1,…,n

A straightforward algorithm

```
Input: X (n obs., p variables), K #groups

Initialize K medoids Mk

REPEAT

    Assignment. Assign each observation to the
    group with the nearest medoid

    Update. Recalculate the medoids from
    individuals attached to the groups

UNTIL Convergence

Output: A partition of the instances in K
groups characterized by their medoids Mk
```

May be K instances selected randomly. Or, K instances which are nearest to the others.

The pairwise distance between the data points being calculated beforehand, it is no longer necessary to access to the database.

Inevitably, the dispersion E_k between the cluster C_k decreases (at least remains stable)

Fixed number of iterations
Or when E no longer decreases
Or when the medoids M_k are stable

⇨ The process minimizes implicitly the overall measure **E**

⇨ The complexity of this approach is especially dissuasive !

PAM Algorithm

Partitioning around medoid (Kaufman & Rousseeuw, 1987)

Here again, it needs to calculate the matrix of pairwise distance d(i,i').

```
Input: X (n obs., p variables), K #groups

Initialize K medoids Mk

REPEAT

    Assign each observation to the group with
    the nearest medoid

    For Each medoid Mk

        Select randomly a non-medoid data
        point i

        Check if the criterion E decreases if
        we swap their role. If YES, the data
        point i becomes the medoid Mk of the
        cluster Ck

UNTIL The criterion E does not decrease

Output: A partition of the instances in K groups
characterized by their medoids Mk
```

K data points selected randomly

BUILD Phase

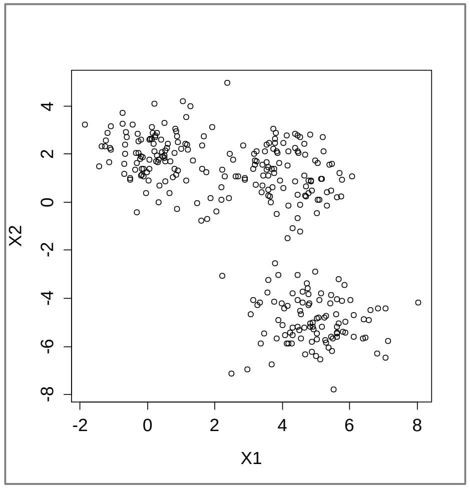
SWAP Phase

See a step by step example on https://en.wikipedia.org/wiki/K-medoids

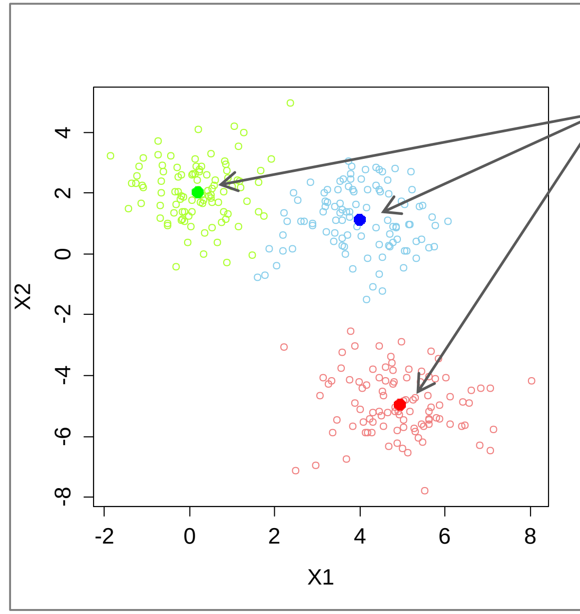
The complexity of the approach remains excessive

PAM Algorithm

PAM vs. K-Means on an artificial dataset



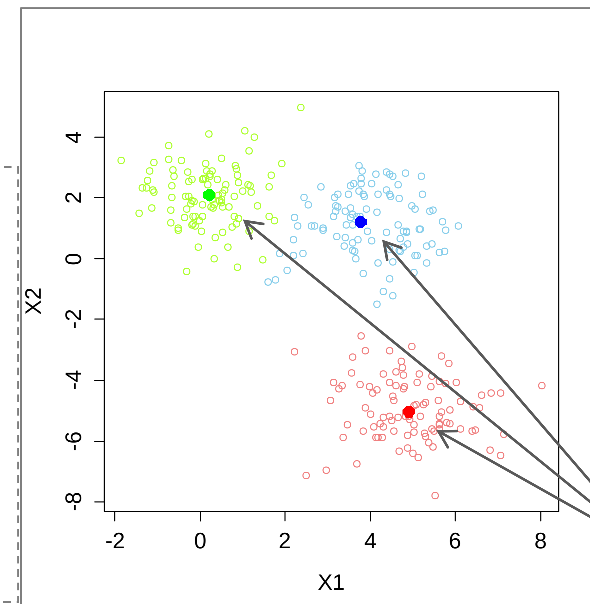
Centroids of the clusters



K-Means

```
> library(cluster)
> res <- pam(X,3,FALSE,"euclidean")
> print(res)

> plot(X[,1],X[,2],type="p",xlab="X1",
ylab="X2",col=c("lightcoral","skyblue","greenyel
low")[res$clustering])
> points(res$medoids[,1],res$medoids[,2],
cex=1.5,pch=16,col=c("red","blue","green")[1:3])
```



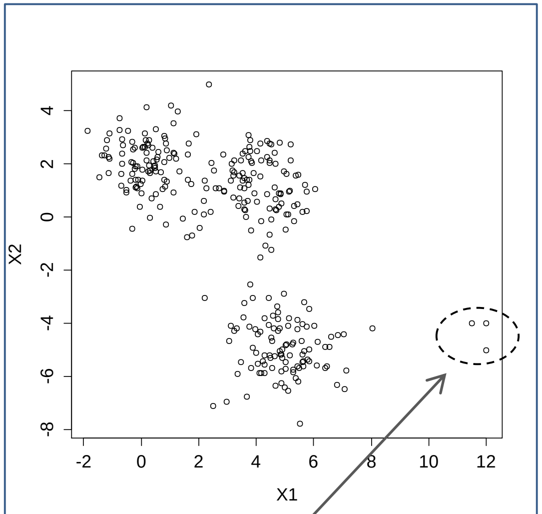
PAM

Because the shapes of the clusters are spherical, the medoids are almost equivalent to the centroids.

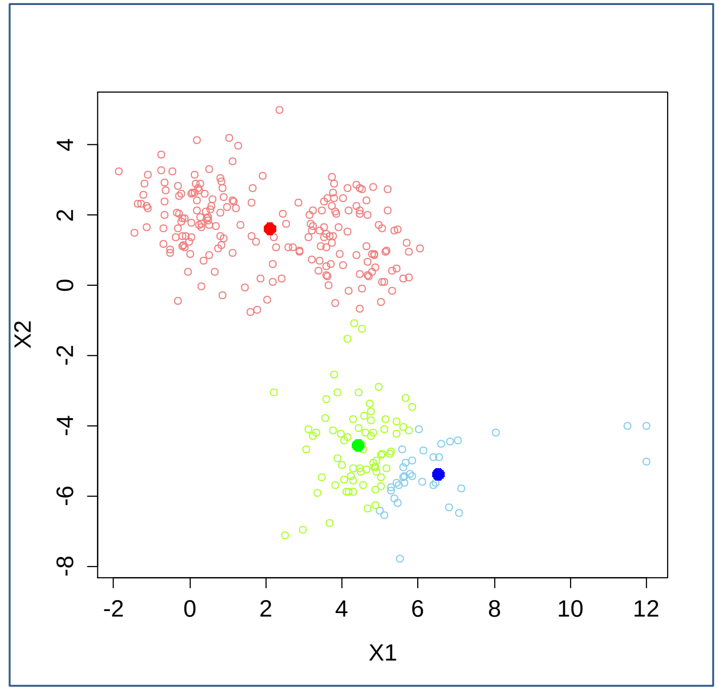
Medoids of clusters

PAM Algorithm

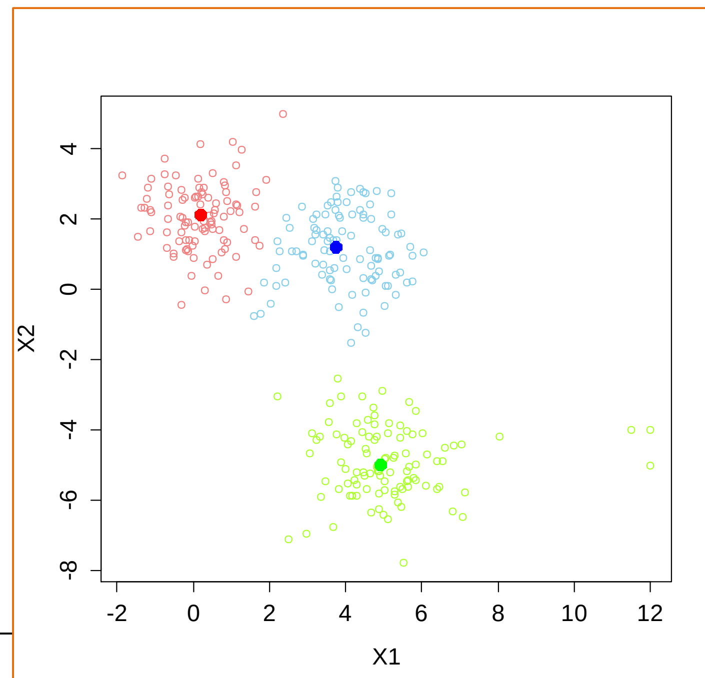
PAM vs. K-Means on an artificial dataset with outliers



Outliers. Source of trouble typically.



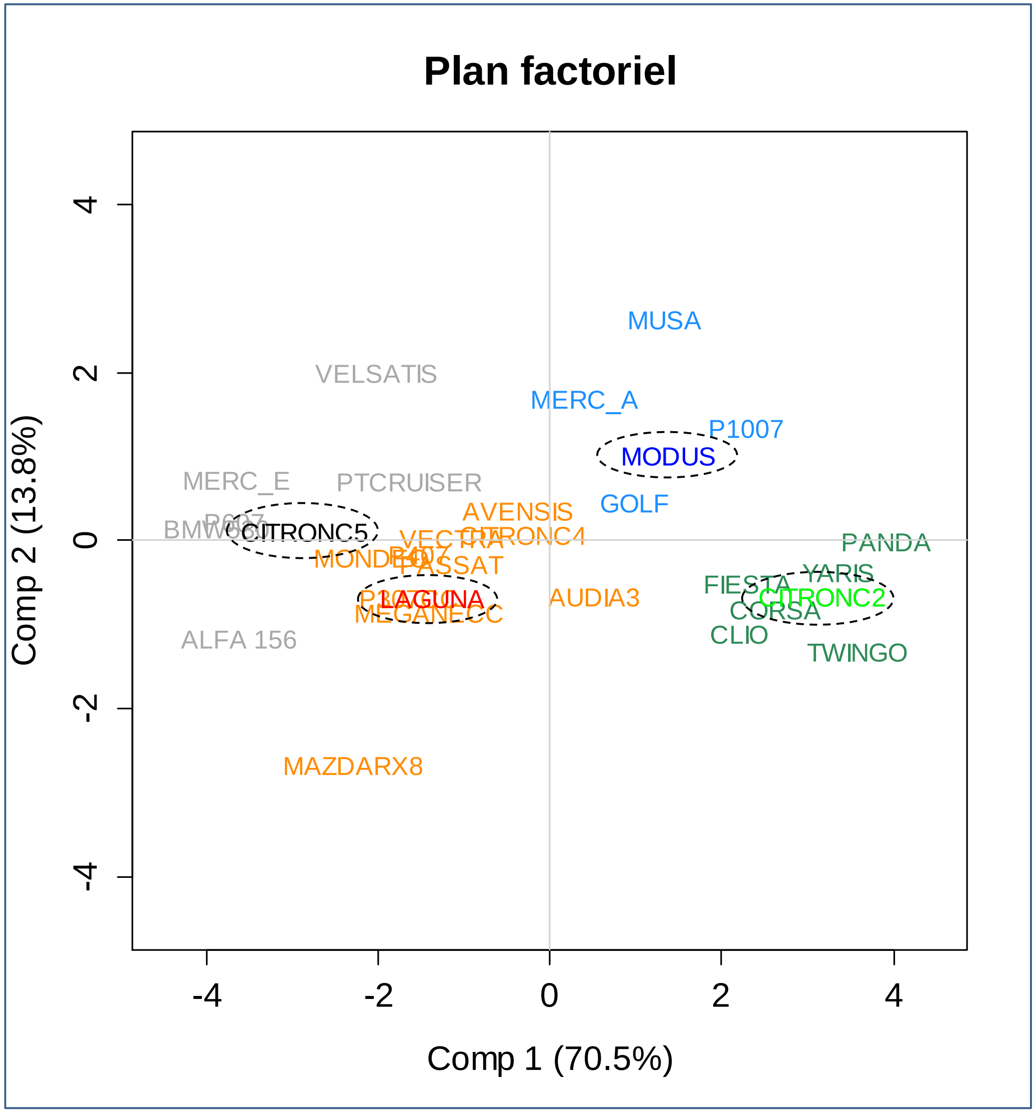
K-Means can be distorted.



PAM remains valid. The medoids are placed wisely.

PAM Algorithm

PAM on the Cars Dataset



PAM

Plotting the instances into the individuals factor map (principal components analysis). We distinguish the clusters. The medoids are highlighted (dotted circle).

CLARA Algorithm

Clustering Large Applications
(Kaufman & Rousseeuw, 1990)

CLARA extends their k-medoids approach for a large number of objects. It works by clustering a sample from the dataset and then assigns all objects in the dataset to these clusters.

```
Input: X (n obs., p variables), K #clusters

Draw S samples of size η (η << n)

Apply PAM algorithm on each sample ➔ S
vectors of medoids

For Each vector of medoids

    Assign all the instances to its cluster

    Evaluate the quality of the partition E

Retain the solution which minimize E

Output: A partition of the instances in K groups
characterized by their medoids Mk
```

In practice: S = 5 and η = 40 +2 x K are adequate [*default settings for clara() in the R "cluster" package for Cluster Analysis*].

Only one single pass on the data is sufficient to evaluate all the configurations.

➔ Ability to process large databases

➔ The algorithm is heavily dependent on the size and representativeness of the samples

CLARA Algorithm

Example for the “waveform” dataset (Breiman and al., 1984)

This is an artificial dataset. The “true” class (CLASSE) membership of the individuals is known.

21 descriptors

V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	CLASSE
-0.29	-2.24	-0.65	0.73	-0.15	1.37	3.21	1.31	1.94	2.06	1.86	2.67	1.42	4.4	3.99	4.17	1.57	1.28	2.34	2.32	-1.17	A
-1.82	1.89	1.1	0.76	2.42	4.59	3.54	3.55	3.07	4.81	1.74	1.65	2.31	2.64	2.73	2.2	1.85	0.33	0.04	-0.85	1.03	A
1.11	-0.42	1.4	-0.27	0.12	2.35	5.86	3.73	4.42	3.72	2.67	2.27	0.44	1.58	-0.02	2.48	0.58	1.04	0.46	1.55	-0.39	B
-1.57	0.52	0.55	1.67	4.56	2.15	0.04	5.24	2.94	1.15	0.48	1.64	0.2	0.26	1.37	3.03	2.03	1.28	0.53	1.07	0.23	A
-0.72	-0.44	-1.02	-0.49	-0.63	0.92	2.42	2.81	4.03	4.33	6.45	5.84	3.88	3.77	1.41	1.32	0.06	-1.22	0.28	-1.65	-0.42	C

30.000 obs.

The two partitions are almost equivalent. But, the computation time is dramatically reduced with CLARA.

PAM : 443 sec. (+ de 7 min)

CLARA : 0.04 sec.

		CLARA		
		C1	C2	C3
PAM	A	9362	485	249
	B	2	9147	1277
	C	852	153	8473

Cramer’s V = 0.85

With the external validation (knowing the real class membership), the various approaches provide similar performances.

Crosstab between CLUSTER vs. CLASSE.

Cramer’s V. PAM, CLARA, K-Means ≈ 0.5

The three methods encounter the same difficulties on this dataset.

Silhouette analysis

A tool for selecting the number of clusters

Silhouette criterion

How well the object lies within its cluster

Rousseeuw (1987) provides a criterion which enables to evaluate a partition independently to the number of clusters (Silhouette).

$$a(i) = \frac{1}{n_a - 1} \sum_{\substack{i'=1 \\ i' \neq i}}^{n_a} d(i,i')$$

Average distance of a data point i with all the other data within the same cluster C_a of size n_a.

$$d(i,C_k) = \frac{1}{n_k} \sum_{i'=1}^{n_k} d(i,i')$$

Average distance of data point i with all the instances of the cluster C_k – other than C_a – of size n_k.

$$b(i) = \min_{k \neq a} d(i,C_k)$$

Distance to the nearest cluster in the sense of $d(i,C_k)$

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

Level of membership to its cluster of the individual i, by comparing the distance to its cluster with the distance to the nearest cluster. s(i) it is independent of K - the number of clusters - because we consider only the distance to the nearest cluster!

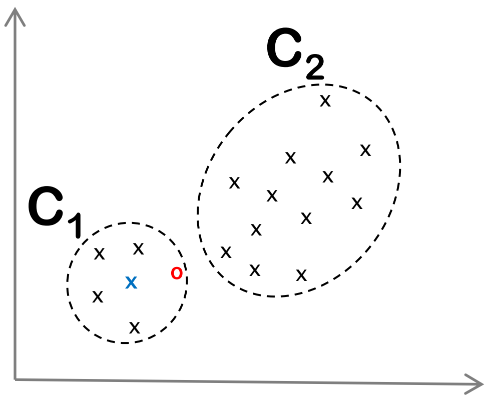
s(i) → 1 : the data point is well positioned within its cluster

s(i) ≈ 0 : the individual is very close to the decision boundary between two neighboring clusters

s(i) → -1 : the data point might be assigned to the wrong cluster

Silhouette criterion

Evaluation of the cluster and the partition



s(x) > s(o): (1) because « x » is near the central position (it is the medoid of the cluster) into C_1 ; (2) because « o » is closer to the cluster C_2.

$$\bar{s}_k = \frac{1}{n_k} \sum_{i \in C_k} s(i)$$

Characterize both the cohesion of the cluster C_k and its separation to the other clusters.

$$S_K = \frac{1}{n} \sum_{k=1}^{K} n_k \times \bar{s}_k$$

Average silhouette. Characterize the overall quality of the partition in K groups. As a rule of thumb:

S ∈ [0,71 ; 1] : strong separation

S ∈ [0,51 ; 0.70] : medium separation

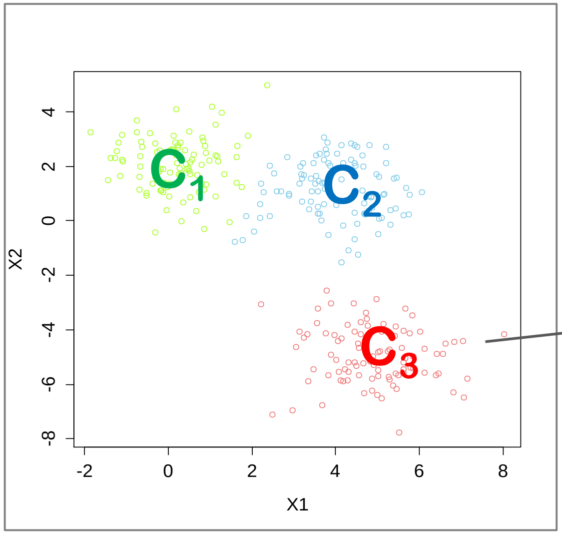
S ∈ [0,26 ; 0.50] : low separation, may be questionable

S ∈ [0 ; 0.25] : the partition seems not meaningful

Silhouette criterion

A tool for determining the number of clusters

Determining the number of clusters is an open problem in cluster analysis. The silhouette criterion being independent to the number of clusters, we can choose the value K which maximize the criterion.



$$\bar{s}_1 = 0.60$$

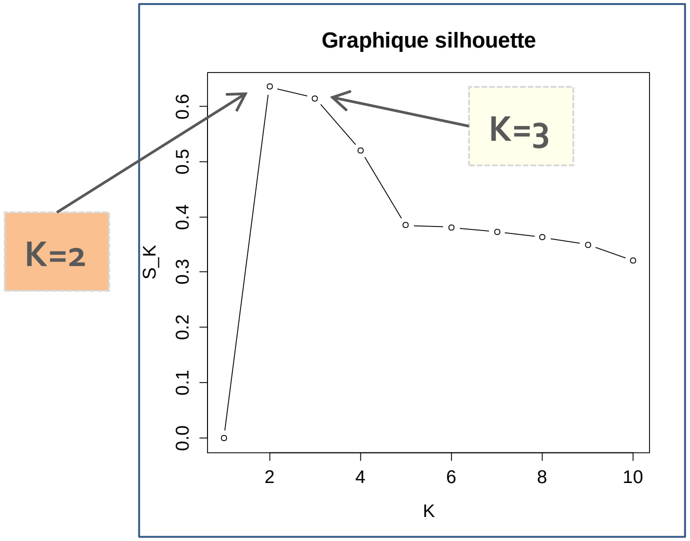
$$\bar{s}_2 = 0.53$$

$$\bar{s}_3 = 0.70$$

The cluster C3 is the one which is furthest to the others.

$$S_{K=3} = 0.61$$

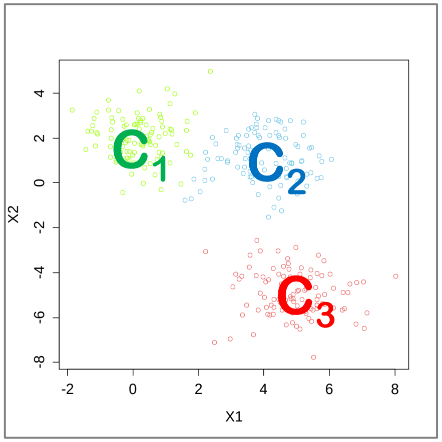
Overall quality of the partition in K = 3 groups.



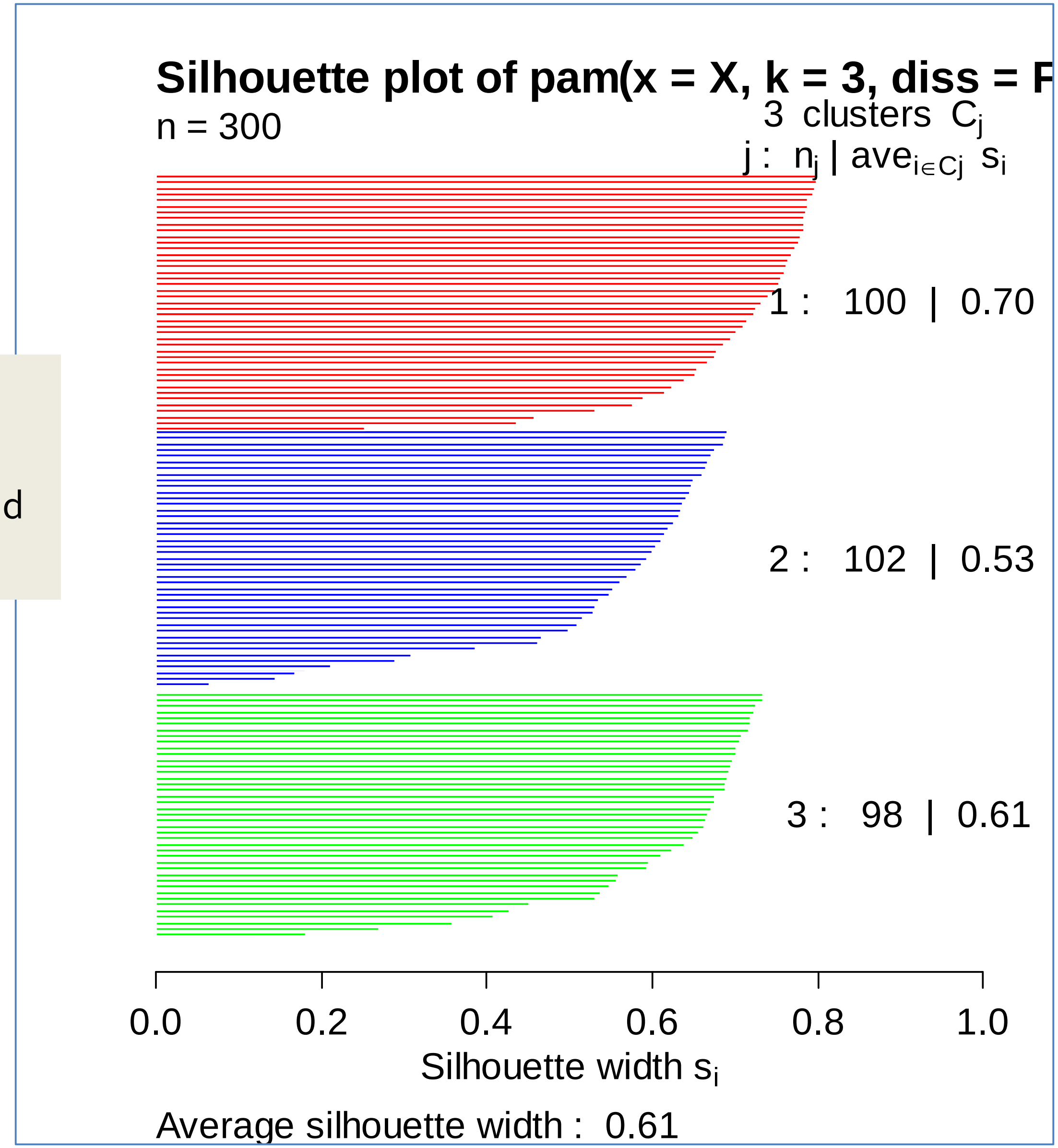
Try various values of K and identify the best solution(partition in K clusters). Here K= 2 (S2 = 0.63) and K = 3 (S3 = 0.61) are competing. *Why the solution K = 2 clusters appears always as the best one whatever the criteria used?*

Silhouette plot

Evaluating clusters



Some popular tools provide a graphical representation called "silhouette plot".



We observe on the one hand the cohesion of the cluster (the group has a higher value s_k than the others); on the other hand, we observe the homogeneity of the situations within the cluster. *For instance, for the red group, only few instances have a low value of silhouette s(i).*

Possible extensions

- The algorithm can be applied to a dataset with categorical variables (e.g. using the chi-squared distance).
- The tandem analysis approach (factor analysis + clustering) is also possible, we can process dataset with mixed data (with both numeric and categorical variables).
- Extension to fuzzy clustering is possible ("fanny" algorithm).

- The extension to the clustering of variables is also easy: r^2 can be used as similarity measure between variables, $(1 - r^2)$ is the distance measure [or respectively r and $(1-r)$ if we want to take account the sign of the association].

Conclusion

- Partitioning methods are often simple. This is an advantage.

- The k-medoids approaches enables to alleviate the problem of outliers, by modifying the notion of representative points of the clusters, by using also appropriate distance measure (e.g. Manhattan distance).
- PAM (Partitioning Aroung Medoids) is a popular implementation of the approach. But the necessity to calculate distances between pairs of individuals is very expensive in computation time.
- We can dramatically improve the ability to handle large datasets by working on samples (CLARA method).

- A criterion for evaluating the partitions, insensitive to the number of clusters, is proposed: the criterion silhouette.
- We can use it to determine the right number of clusters. But it is a heuristic, the choice of the number of clusters must be supported by the interpretation.

References

Books and articles

Gan G., Ma C., Wu J., « Data Clustering – Theory, Algorithms and Applications », SIAM, 2007.

R. Rakotomalala, « k-means clustering », octobre 2016.

Struyf A., Hubert M., Rousseeuw P., « Clustering in an Object-Oriented Environnement », Journal of Statistical Software, 1(4), 1997.

Wikipedia, « k-medoids ».

Wikipedia, « Silhouette (clustering) ».